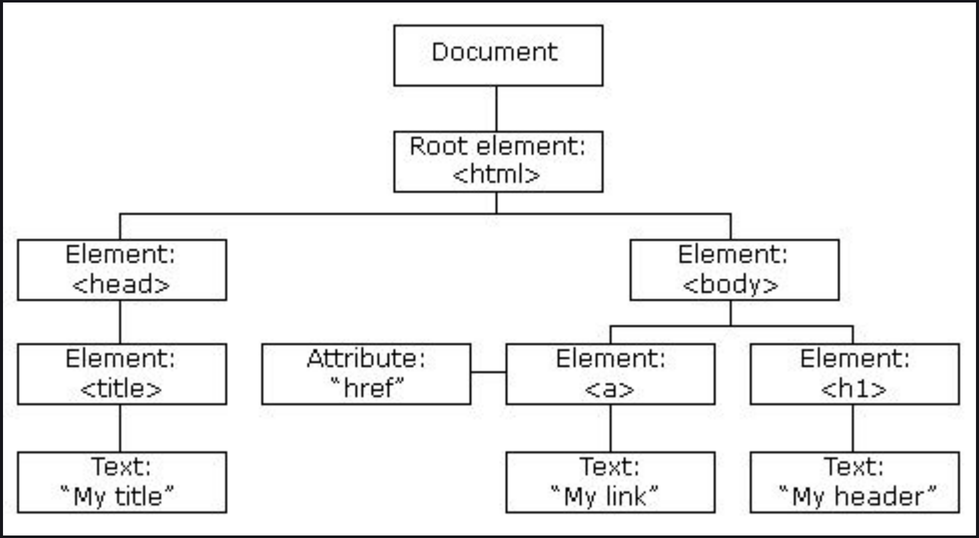
MISC

Cross-Site Scripting (XSS)

WHAT MAKES A WEBSITE?

What is a DOM?

Document Object Model represents document page as nodes that can be accessed and altered by JavaScript



javascript has access to
html documents via DOM api

alter innerHTML

```
<div class = “header”>
```

join misc at umisc.info <3

```
</div>
```

javascript has access to
html documents via DOM api

alter innerHTML

```
<div class = “header”>

        pls join xoxo

</div>
```

javascript has access to
html documents via DOM api

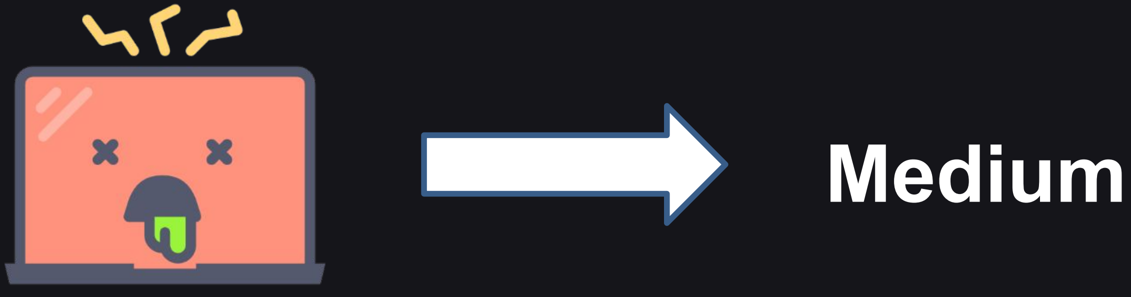
```
<div class = “header”>

        pls join xoxo

</div>
```

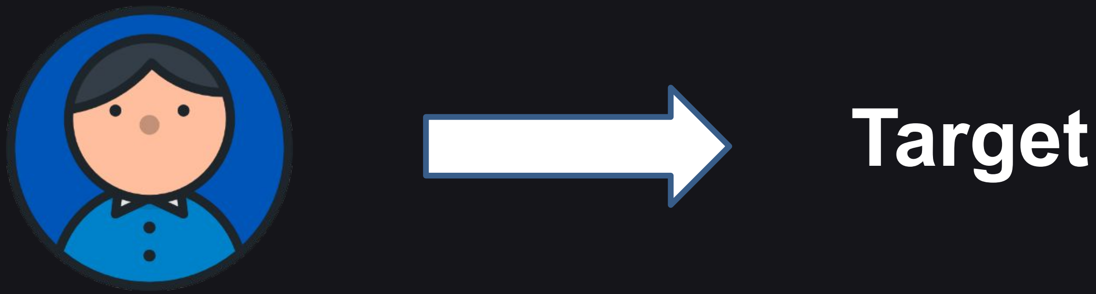
document.cookie

Can we inject JavaScript into a website?

So… what is XSS?

- **Form of code injection (Typically JavaScript)**
- **Vulnerable web applications are used to exploit users**





HOW DOES XSS WORK?

1. **Websites and web apps have multiple channels to take user input**

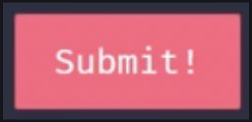
2. **Vulnerable web apps do not process user inputs securely**

3. **Malicious instructions (scripts) can be passed**

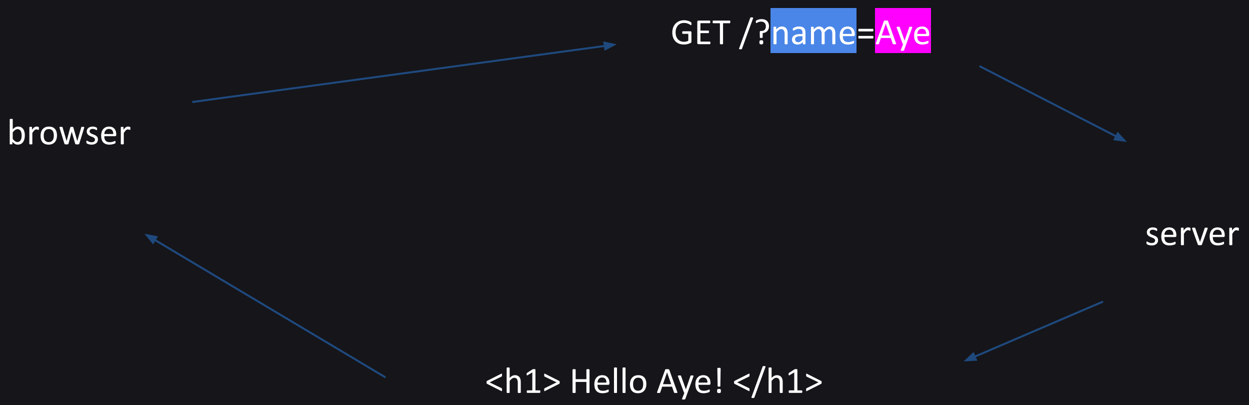
4. **The vulnerable application processes these scripts**

Hello user, what is your name?

Aye



Hello Aye!

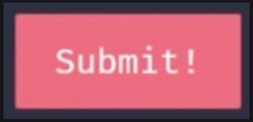


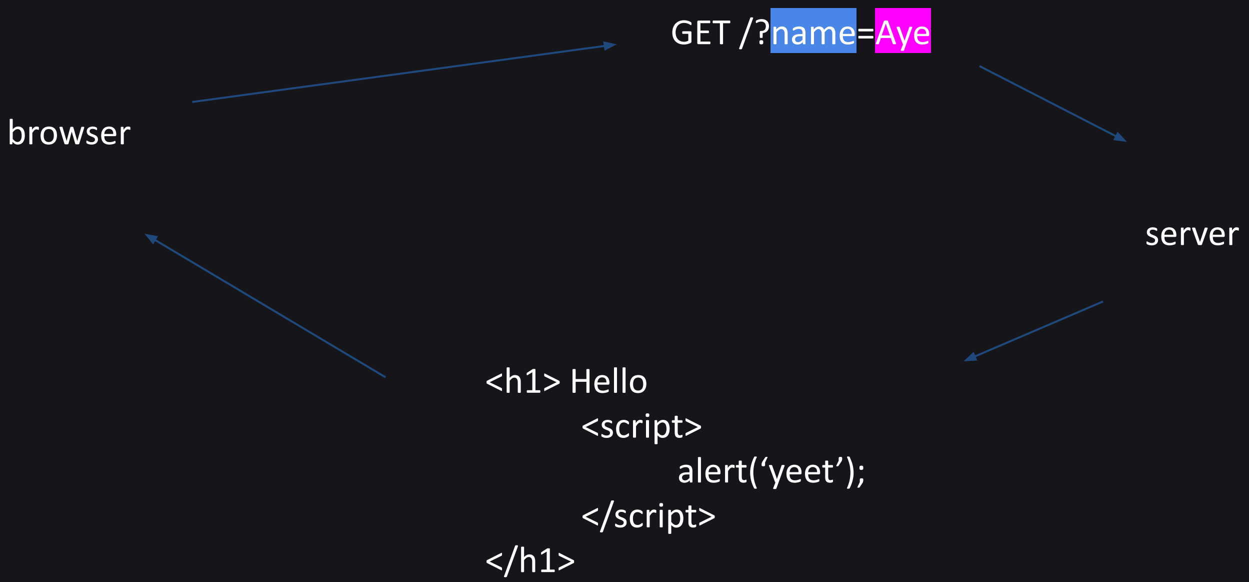
```
GET /?name=Aye
```

```
<h1> Hello Aye! </h1>
```

Hello user, what is your name?

<script> alert(‘yeet’) **</script>**





```
<h1> Hello
        <script>
                alert('yeet');
        </script>
</h1>
```

YEET

WHAT DAMAGE CAN XSS DO?

- Taking ownership of user accounts – session hijacking, stealing credentials
- Defacing websites
- Injecting Malwares
- Inducing user action – Make it look the victim has done it
- Exploiting trust relations

TYPES OF XSS

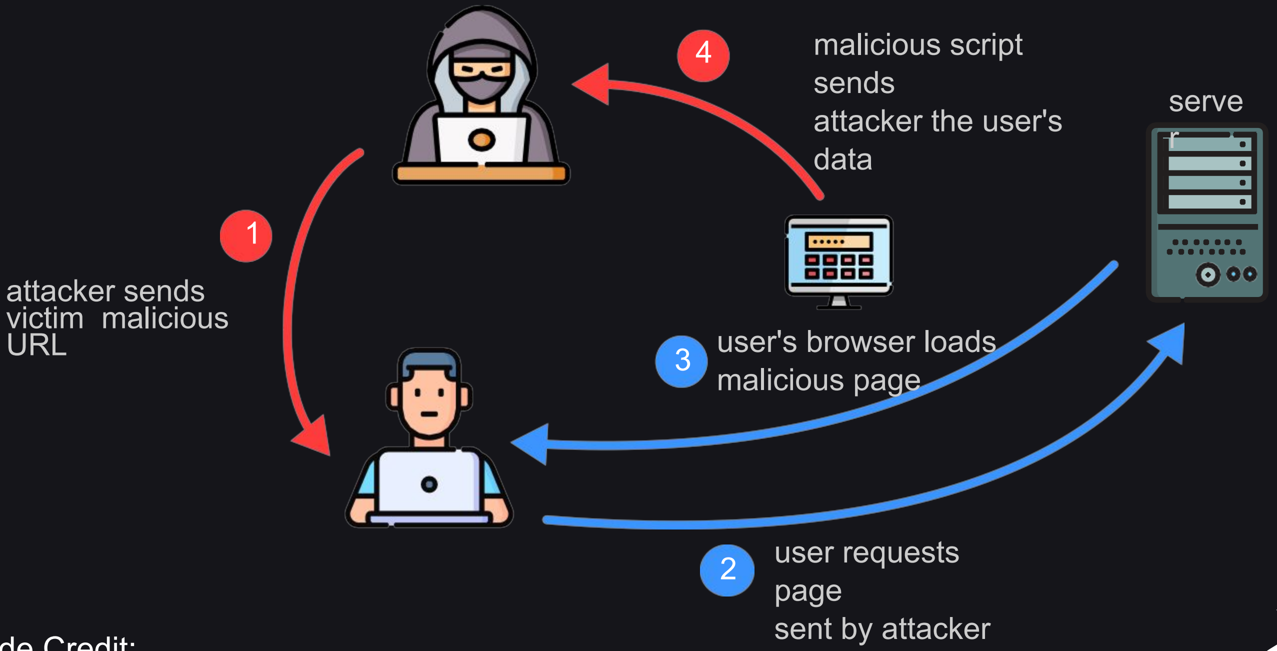
There are three main types of XSS

1. **Reflected XSS**
2. **Stored XSS**
3. **DOM Based XSS**

Reflected and stored XSS is still very common. In fact
XSS is responsible over 70% of web vulnerabilities!

REFLECTED XSS

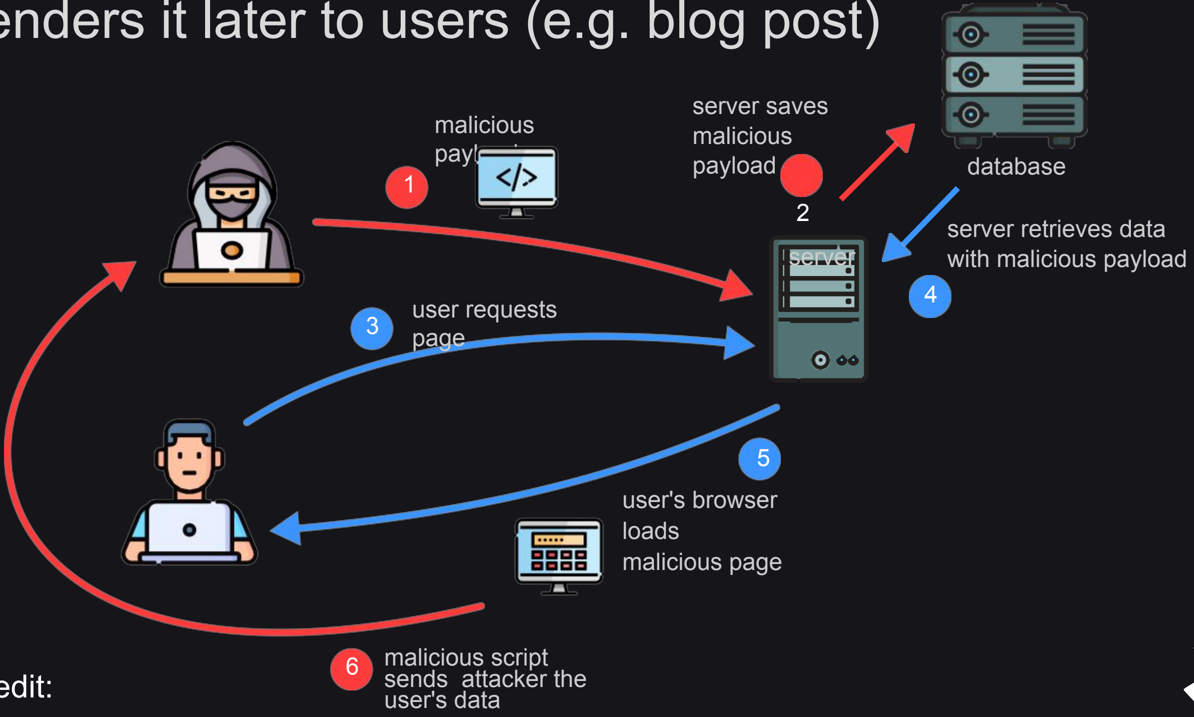
- Occurs when unsanitised user input is displayed in the webpage


4
malicious script sends attacker the user's data
serve r
1
attacker sends victim malicious URL
3
user's browser loads malicious page
2
user requests page sent by attacker

Slide Credit: Joseph

STORED XSS

- Occurs when a web app saves user input to a database and
  renders it later to users (e.g. blog post)



Slide Credit:
Joseph

DOM BASED XSS (SELF XSS)

- **The script is run inside victim’s browser**
- **Requires a lot of social engineering to convince the victim**
- **Usually a not a vulnerability anymore as modern
browsers have built-in protection
against running
‘outside’ scripts**

HACK STEPS

1. Choose an unique arbitrary string that does not appear anywhere within the target (‘mytestxssdsdf’)
2. Submit the string at every input parameter of the target
3. Monitor applications responses for every appearance of this string
4. Test HTTP request Methods (GET and POST)
5. In addition to standard request parameters, test instances where application processes HTTP request headers. (‘Referer’ and ‘User-Agent’ are useful ones)

TESTING REFLECTIONS

Example 1: A Tag attribute value

```
<input type=”text” name=”address1” value=”myxsstestdmqlwp”>
```

Exploit:

```
“><script>alert(1)</script>
```

Example 2: A JavaScript String

```
<script>var a = ‘myxsstestdmqlwp’; var b = 123; ... </script>
```

Exploit:

```
‘; alert(1); var foo=’
```

HANDY TOOLS

- xSs jAvAsCrIpT PoLyGloTs
  https://github.com/0xsobky/HackVault/wiki/Unleashing-an-Ultimate-XSS-Polyglot

- Firefox / Chrome Developer Tools (Watch the following video):
  https://www.youtube.com/watch?v=FTeE3OrTNoA

- Burp Suite (Video by legendary Jason Haddix himself!):
  https://www.youtube.com/watch?v=h2duGBZLEek&t=2072s

EPIC RESOURCES

- The Web Application Hackers Handbook (Chapter 12)
- PortSwigger Web Academy (Free)

PLACES TO PRACTICE WITHOUT GETTING ARRESTED

- MISC CTF
- Google Firing Range
- Google XSS Game (https://xss-game.appspot.com/)
- Pentester Lab!

THANK YOU!

Please ask any questions you have in the chat!
Slide credit: Kaif and Joseph, thanks for your help!